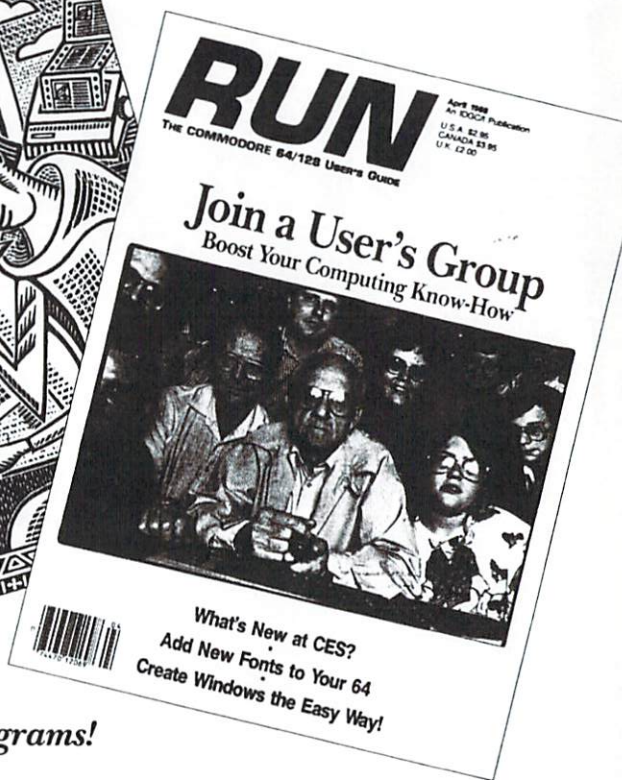
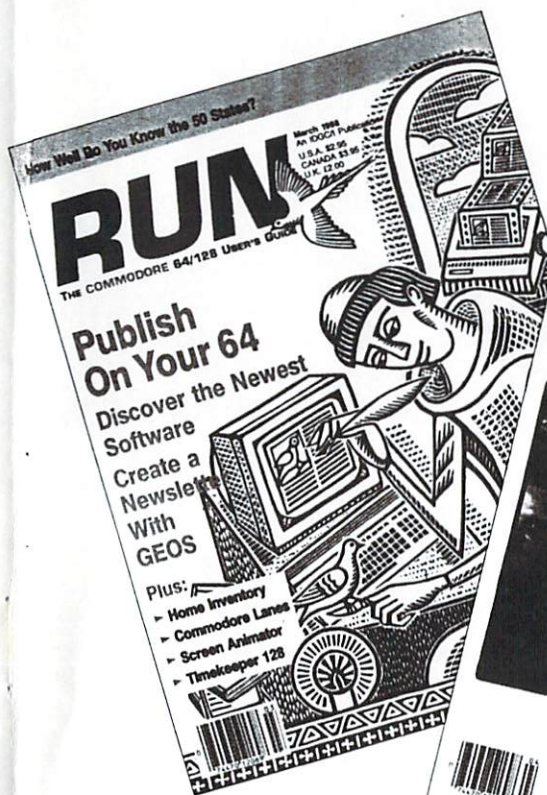


March/April 1988 Edition

RE **== RUN**

RUN Programs on Disk

For the C-64 and C-128



Plus: Extra Bonus Programs!

12 Programs Included on this Disk

Home Inventory ▶ Bowling ▶ 128 Clock/Calendar
Window Construction Set ▶ Fontastic 64 ▶ Mr. Poster
Cursor Commotion 128 ▶ Video Flashcards

From the March RUN:

- ▶ Home Inventory
- ▶ Affairs of State
- ▶ Commodore Lanes
- ▶ Time-Keeper 128
- ▶ Display Frame Animator

From the April RUN:

- ▶ Window Construction Set
- ▶ Fontastic 64
- ▶ Creating Windows
- ▶ Cursor Commotion 128

Plus: Extra Bonus Programs!

- ▶ Video Flashcards
- ▶ 64 Command Collector
- ▶ Mister Poster

If any manufacturing defect becomes apparent, the defective disk will be replaced free of charge if returned by prepaid mail within 30 days of purchase. Send it, with a letter specifying the defect, to:

ReRUN • 80 Elm Street • Peterborough, NH 03458

Replacements will not be made if the disk has been altered, repaired or misused through negligence, or if it shows signs of excessive wear or is damaged by equipment.

The programs in ReRUN are taken directly from listings prepared to accompany articles in *RUN* magazine. They will not run under all system configurations. Use the RUN It Right information included with each article as your guide.

The entire contents are copyrighted 1988 by IDG Communications/Peterborough. Unauthorized duplication is a violation of applicable laws.

©Copyright 1988 IDG Communications/Peterborough



Introduction

March-April '88 ReRUN

LOOKING OVER THE PROGRAMS in this edition of ReRUN, no one theme seems obvious. It may appear that we've taken a liking to windows, with windowing programs for both the C-64 and C-128. On the other hand, you might say this edition leans toward entertainment in offering educational games and a bowling game. Yet, practicality is present with an inventory program, a batch processor and a handy clock/calendar. So, there's a wide variety of programs.

Without further ado, here's a description of what's on this disk. From our March issue of *RUN*, we're presenting Home Inventory. Offered in both C-64 and C-128 formats, this program makes a detailed list of your household belongings for insurance purposes. If you live in California and your home suddenly erupts in flames due to a little seismic activity, a previously created inventory of your valuables and furnishings could avoid a lot of headaches when you try to justify the losses to your insurance company.

Next on the list of March programs is Affairs of State, an educational program that helps people of all ages learn the capitals of all 50 states, plus the official bird, flower and nickname of each.

On the entertainment side, we've got Commodore Lanes, a C-64 bowling game that allows you to play alone or with up to seven of your bowling buddies. The program offers ten levels of difficulty so that handicaps can be used to balance the degrees of skill.

Time Keeper 128, our March Easy Applications program, is an interrupt-driven clock/calendar for the C-128 that operates in 80-Column mode. You'll quickly discover that it's compatible with many machine language programs, including RUN Script 128, versions 2.40 and 2.56.

March's Mega-Magic, Display Frame Animator, is from our well-known Magic contributor, Joseph Charnetski. Joe has created a dazzling title animator that works by rapidly cycling through character colors. You can use it to create glowing, neon-like borders around titles in any program. It works in both 40- and 80-Column 128 mode, as well as in 64 mode.

Moving on to the April issue, we're offering Window Construction

Set. For many long-time *RUN* readers and ReRUN users, John Ryan's name is synonymous with high-quality machine language programs, such as *RUN*'s now-famous 64 DOS Shell. John has now produced a sophisticated, interrupt-driven windowing system for the C-64 that allows the simultaneous display of up to nine windows. Since it's a memory-resident utility that "hides" until summoned, it can be used concurrently with Basic programs.

Everybody loves new C-64 character fonts, right? With that in mind, I'm sure you'll be excited to hear that Fontastic 64 on this edition of ReRUN spices up your screen displays by adding nine dynamic new fonts.

Our April Easy Applications program, Creating Windows, is for the C-128's 80-Column mode, but can be used in 40-Column mode with one line change. Primarily, it gives you insight and exposure to the C-128's useful Window command.

Also from the April issue comes one of my favorite Mega-Magic programs, Cursor Commotion 128. Written by *RUN*'s former associate editor, Jim Borden, this program is a 40- and 80-Column C-128 version of the immensely popular Cursor Commotion for the C-64 that we published in last September's Mega-Magic column. You'll be able to spend hours making animated designs and messages, saving them to disk, and then playing them back again. As with the C-64 version, I'm placing an 80-Column Cursor Commotion demo on this edition of ReRUN.

Now for our bonus programs. First, Video Flashcards, an educational game for children, allows parents to drill their children with addition and subtraction problems. The second program, 64 Command Collector, allows you to convert your C-64 into a batch processor. The third bonus is Mister Poster, a versatile graphics expander that allows you to blow up a graphics image, such as a Doodle! picture, to poster size. By far the gentlest poster maker I've ever used, Mister Poster does not beat your printer to death and burn up ribbons in the process of printing a poster.

As always, I want to express my thanks to all of you for your interest in ReRUN. I'll look forward to bringing you the next edition.



Technical Editor
RUN magazine

Directory

PAGE	DOCUMENTATION	DISK FILENAME	FILE TYPE
		* MENU 128 _____	BASIC
		MENU 64 _____	BASIC
2	HOME INVENTORY	HOME INV. 64 _____	BASIC
		* HOME INV 198 _____	BASIC
7	AFFAIRS OF STATE	AFFAIRS OF STATE _____	BASIC
		AFFAIRS 2 _____	BASIC
7	COMMODORE LANES	BOWLING 64 _____	BASIC
8	TIME KEEPER 128	* TIME KEEPER 128 _____	BASIC
		BAS.TK128 _____	BASIC
		OB.TK 128 _____	ML
11	DISPLAY FRAME ANIMATOR	DISPLAY ANIMATE _____	BASIC
12	WINDOW CONSTRUCTION SET	WINDOW CONSTRUCTN _____	BASIC
		WINDOWS LIST2 _____	BASIC
		WCS _____	ML
		WCS FILE _____	BASIC
15	FONTASTIC 64	BOOT CHAR SET _____	BASIC
		BLOCK SET _____	BASIC
		CAPS SET _____	BASIC
		COMPUSSET _____	BASIC
		CURSIVE SET _____	BASIC
		DIAMOND SET _____	BASIC
		DOT SET _____	BASIC
		STENCIL SET _____	BASIC
		THIN SET _____	BASIC
		UNDER SET _____	BASIC
		CHAR SET GENRTR _____	BASIC
		FONT MENU _____	BASIC
		BLOCKFONT _____	ML
		CAPSFONT _____	ML
		COMPUFONT _____	ML
		CURSIFONT _____	ML
		DIAMONDFONT _____	ML
		DOTFONT _____	ML
		STENCILFONT _____	ML
		THINFONT _____	ML
		UNDERFONT _____	ML

PAGE	DOCUMENTATION	DISK FILENAME	FILE TYPE
17	CREATING WINDOWS	* C-128 WINDOWS _____	BASIC
19	CURSOR COMMOTION 128	* CRSR COMM. 128 _____	BASIC
		* CRSR COMM. DEMO _____	BASIC
19	VIDEO FLASHCARDS	£ VIDEO FLASHCARDS _____	BASIC
21	C-64 COMMAND COLLECTOR	£ COMMAND COLLECT _____	BASIC
24	MISTER POSTER	£ MR.POSTER.RUN _____	ML
		£ MR.POSTER.GEN _____	BASIC
		£ DDC-64.RUN _____	ML

* - C-28 mode only

£ - Bonus program

Before you run a program, read the documentation that pertains to it.

How To Load

LOADING FROM MENU

To get started, C-64 users should type LOAD "MENU 64",8 and press the return key. When you get the Ready prompt, the menu is loaded and you should type RUN to see a list of the programs on your disk. C-128 users need only press the shift and run/stop keys. When all the programs are displayed on the screen, you can run the one you select by pressing a single key.

LOADING FROM KEYBOARD

If you do not wish to use the menu program, follow these instructions.

C-64: To load a C-64 program written in Basic, type: LOAD "DISK FILENAME",8 and then press the return key. The drive will whirl while the screen prints LOADING and then READY, with a flashing cursor beneath. Type RUN and press the return key. The program will then start running. To load a C-64 program written in machine language (ML), type: LOAD "DISK FILENAME",8,1

C-128: All C-64 programs can be run on the C-128 as long as your computer is in C-64 mode. All C-128 programs are clearly labeled on the directory page. Your C-128 *must* be in C-128 mode to run these programs. To load a C-128 mode program, press the F2 key, type the disk filename and then press the return key. When the program has loaded, type RUN.

MAKING COPIES OF ReRUN DISKS

Many programs on your ReRUN disk have routines that require a separate disk onto which the program writes or saves subfiles. To use these programs, you must first make a copy of the original program onto another disk that has enough free space on it to hold these newly written subfiles.

It's simple to make a copy of a Basic program. Just load it into your computer as outlined above, and then save the program back onto a separate disk that has plenty of free space for extra files.

Copying an ML program is not so simple. You cannot simply load and save an ML program; you'll need to use a disk-backup utility program, such as the one on your Commodore Test Demo disk.

RUN it right: C-64; C-128; printer optional

Home Inventory

By Ian Adam

WHEN FIRE OR THEFT STRIKES YOUR HOME, nothing can make up for the feelings of disruption and violation of privacy it evokes, but at least, you reassure yourself, insurance will help cover the financial loss. However, if you've ever made a claim for lost property, you know what a frustrating experience it can be. Even if you can recall everything that was lost and describe it in some detail, you're still likely to have arguments over quality, condition and such. What's more, without hard identification such as serial numbers and dimensions, it's nearly impossible for the police to recover and return belongings.

You can avoid a lot of these problems by using my Home Inventory program to prepare a thorough list of your possessions in advance. Such a list can help you decide how much insurance you need, then speed and strengthen any claim you may have to make, as well as assist the police in recovering your possessions if theft is involved.

TWO VERSIONS

Home Inventory consists of programs for the C-64 and C-128. I'd suggest you use a work disk with this application, containing the program itself and the inventory files you'll input. In that way, everything will be handy when you want to work on your inventory.

The C-128 version has a number of more sophisticated features than the 64 version. First, it supports the 80-column screen, allowing you to see more information while you type. Second, it uses windows for displaying instructions and inventory status.

STARTING A FILE

When you run Home Inventory, the first screen display lists the status of your file, including the number of rooms and number of items in each room that you've allowed for, plus the number of items the inventory contains so far. Of course, if you're just starting, these numbers are zero. The main menu appears at the bottom.

Choose option 1, Create File; then, at the prompt, supply a name for it that's no more than 12 characters long. I'd suggest using the date, such as MARCH 1988, as a filename, so it's easy to find your most recent update on the disk. Whatever name you choose, the program will automatically append .INV to it (e.g., MARCH 1988.INV), to identify it as a Home Inventory file. When you want to load the file later, don't type the .INV. If you've amassed more items than one file can hold, just expand your inventory into a second.

The program assumes that you'll group items according to the room where they're most commonly kept. After naming the file, you must specify how many rooms you want and the maximum number of items per room. The number of each that your computer can handle will depend on how verbose you are, but, as a rough guide, figure the C-64 can handle 300 items (say seven rooms of 40 items each) and the 128 twice that many (a 15-room mansion!).

In some cases, you may want to define rooms that aren't actual physical spaces. For instance, you could call your collection of old records a room, since there might not be space to list all the individual disks under "living room."

After you specify the number of rooms and items, the file-status/main-menu display reappears, with the status lines reflecting the existence of your new file.

If you've already created a file and saved it to disk, you can load it for updating by selecting main menu option 2, Get File from Disk. The program then gives you a chance to change the item limits. To keep the same size file, press the return key at each prompt. If you decide to reduce the file size, be careful, because some data items may be lost.

As you use Home Inventory, keep in mind that the main menu options of creating a new file, loading a file from disk (option 2) and ending the program (option 0) will erase the data you have in memory. For this reason, if you select one of these options, you'll be asked to confirm your intention before the program will execute the option. Assuming you have a copy of the data on disk, press Y and return to tell the program to proceed.

COMPILING YOUR INVENTORY

After you've finished creating a file or loading one from disk, choose main menu option 3, Work on Data, and the program shows you the list of rooms and the number of items described for each.

To select a room to work on, type either its number or defined name. After you select a room, you get a second menu. To add items, choose option 1.

(If you're opening a new room, give it a name, then start entering items at the prompt, thus bypassing the second menu.)

This is where you must be thorough. Scour the real room carefully and enter all items of any value (small items may be grouped as one entry). For each item, enter a complete description, including make and model, size, color, current condition, and so forth. You're allowed up to two screen lines per item—80 characters on the C-64 or 160 characters on the C-128.

The only limitation on your input, since this is Basic, is that you use no commas or colons; I find that semicolons and slashes make quite adequate separators, anyway. You can get around this constraint if you must by starting your input with a quotation mark, but this makes the description more difficult to edit.

Next, enter any unique identifying characteristics, such as serial number, unusual dimensions or scratches, that would help establish the item as yours in court.

The remaining data to be entered includes the approximate age of the item, in years, the price you originally paid for it and its present value. Your insurance company will need this information in handling your claim.

There are two approaches to present value. Traditionally, you'd enter the depreciated value—that is, the amount the item would probably fetch when sold used. An item that's been in use for a few years and is still in fairly good shape might be valued at about half its new cost, for example. If you have a replacement-cost endorsement in your insurance policy, the present value is defined as the cost of a new, comparable item. In your inventory, use whichever method your insurance calls for—just be consistent.

When you're finished with a room, enter Q in place of the item description to return to the main menu. You can always come back later and add more items, until the room is full.

REVIEWING YOUR DATA

If you want to make additions or alterations to data you've already entered, type in the room number or name, then choose the option you want from the data revision menu that then appears on the screen. Option 1 is for adding items, while option 2 displays the descriptions of all items currently in the room. The items scroll by fast, so you may

want to slow or stop the listing with the keys appropriate to your computer (CTRL on the 64 and the Commodore key on the 128). Press return at the end to get back to the revision menu.

To see more details on a specific item, choose option 3 in the revision menu and enter the item number. To get back to the revision menu, enter 0 for the item number.

If you want to change some information on an item, choose the revision menu's option 4, Modify Data, and enter the item number. After the program lists the present data for that item, type in any changes you want to make; press return for data to be left unchanged. When finished, enter 0 as the item number to get back to the revision menu.

Once you've made all the changes you want to the current room, choose option 5 to move to another room or option 0 to return to the main menu.

SAVING TO DISK

While building your inventory, be sure to save the file to a work disk frequently, so you won't lose a lot of work to a computer failure or power glitch. To save an inventory file, choose option 4 from the main menu. When you're asked for the filename, just press return to save the file under its current name. If you want to change the filename, type in the one you want, and the name will be changed accordingly. If you type in a name you've used before on the disk, the program will tell you the file exists and ask if you want to replace it. If so, press Y and return to erase the old file and save the new one in its place. Otherwise, enter N and try again, using a different filename.

PRINTED REPORTS

After you have your inventory on disk, you can use option 5 in the main menu to prepare printed reports of various kinds. When you select this option, the program first asks if you want the printout to start with room 1. If so, press return; if not, press the number of the room you want it to start with. Then the program asks if you want the printout to end with the last room in your file. If so, press return; if not, press the number of the last room you want included. After indicating your choices, you'll get the print menu.

There are many combinations of printers and interfaces in Commodoreland, and I've written Home Inventory to support as many of them as possible. If you encounter difficulties, such as double-

spacing or funny graphics characters, try changing the secondary address, using option 4 in the print menu (see below). The manual for your printer or interface will suggest what value to use to get upper- and lowercase letters, with line feeds.

Home Inventory can generate various kinds of reports. A Summary Data report gives the number of items in each room, their total cost and present value, and the total data for the specified range of rooms. This type of report is helpful for deciding in advance how much insurance you need.

The Short ID, Age and Cost report is a table of short descriptions and full financial data for all items. In building this table, you may have to trim down some descriptions and IDs, because only 46 and 15 characters, respectively, are allowed for them. Don't worry; brief descriptions are plenty for the aim of this report, which is to help you assess your detailed insurance needs and review your growing estate.

When you select the Full Descriptions and IDs report, financial data on individual items is left out and only the detailed descriptions, identifying characteristics and room summaries are printed. This report is for the police to use in identifying and recovering your possessions.

A Full Report includes the full description and ID for each item, along with its age and financial data. This printout allows two lines for long descriptions and IDs. This is the most complete printout, and is the one you should save for your own records.

When your printer has finished, the program returns to the main menu.

FINAL TIPS

When you've completed your inventory, make a backup, including both the program and your data files, as an archival copy. In addition, order up a printout of the full report, put it in an envelope, label it and store it in a safe place, away from home. It won't do you any good if it's been burned in the fire or stolen in the burglary. Good locations include a safe deposit box or your lawyer's or insurance agent's office.

You may be tempted to put off preparing a home inventory, because it takes time and a good deal of concentration. However, let me assure you, it's a lot more difficult to do when you don't have the items in front of you. Use this program to compile your home inventory soon!

Affairs of State

By Mary E. Wilson

EVERYONE SURELY KNOWS that Sacramento is the capital of California. But how many can answer which state claims the black-eyed Susan as its state flower?

This educational program will test your knowledge of not only the state capitals, but also the state flowers, state birds and even the nicknames of all 50 states. You can use it as a tool to help little Bethany pass her seventh-grade social studies quizzes, to test Uncle Bill's trivia knowledge, or simply for your own edification.

After you load and run AFFAIRS OF STATE, a menu will soon appear on the screen, offering a choice of quiz categories. Option 1 asks you to name the capitals; option 2 presents the capitals, and you must correctly name the states. Options 3, 4 and 5 give the state, and you must type in the flower, bird or nickname for that state.

The program automatically keeps track of your score (you get two points for each correct answer). A high-pitched sound accompanies a correct answer. After each wrong entry—announced by a low-pitched sound—the program provides the correct answer.

AFFAIRS 2 will print out a columnar answer sheet, condensed on one page. It will work on the 1525 and compatible printers.

RUN it right: C-64

Commodore Lanes

By John Fedor

BOWLING, A C-64 GAME you'll find yourself playing as intensely as the real thing, lets you and as many as seven of your friends bowl

strikes, spares, gutterballs—anything except splits. You can choose among ten handicapping speeds (the lower the number you select, the faster the ball travels down the lane), and scores are automatically tallied at the end of the game.

The scoresheet, along with each player's name and running score, is displayed at the top of the screen. To roll the ball down the bowling lane, press any key other than the space bar.

Players take turns bowling until all ten frames of the scoresheet are filled. As in the real thing, if you bowl a strike (X) in the tenth frame, you get two more balls to roll, and a spare (/) gives you one extra ball. The highest score possible is 300 points, which requires 12 consecutive strikes.

The game uses two machine language routines. One copies the ROM character set into RAM and adjusts the bytes to resemble a bowling pin. The other draws the bowling ball on the screen and checks for a keypress.

If you think stringing up strikes and spares in Bowling is easier than the real thing, hark to the words of the *RUN* staff, who play the game in the late afternoon, when the editor-in-chief isn't looking: It's almost as difficult.

RUN it right: C-128 (40- or 80-Column mode)

Time-Keeper 128

By Robert Kodadek

TIME-KEEPER 128 IS A HANDY clock/calendar that keeps the time, month, day and year in the upper-right corner of the screen while your computer is running. The program is interrupt-driven, RAM resident and transparent to many other Basic and ML programs.

Time-Keeper works on both 40- and 80-column screens and has a built-in alarm (which you can adjust for visual and/or audible signals) and an autoleap-year function. The clock/calendar is updated once per second and the data is stored in easily accessible RAM locations. This information can then be used by one of your own custom applications, such as a bulletin board system.

RUNNING THE PROGRAMS

Time-Keeper 128 consists of two programs. TIME KEEPER 128 is a boot program to load the object file and set the date, time and alarm. The second, BAS.TK128, creates the fast-loading object file, OB.TK 128, on your disk.

Whenever you want to use Time-Keeper, load and run the boot program. You'll be prompted to enter the correct time and date and whether you wish to set the alarm. Time-Keeper's display begins with an asterisk followed by the month, date, year, hours, minutes, seconds and the characters "al," for alarm.

The leading asterisk prevents the Basic editor from replacing or deleting a Basic program line should you press return while the cursor is resting on the top line of the screen. Whenever the alarm is set, the characters "al" will appear in reverse video. When the alarm goes off, these characters will flash while the SID chip emits a bell-like chime for about 30 seconds. If you don't care to listen to it, cancel the chiming with the shift/control/C combination.

Though the display is continuous, you may turn it off at any time with shift/control/F and back on again with shift/control/N. These key combinations are rarely found in programs, so they shouldn't interfere with any program that's running concurrently with Time-Keeper.

USING TIME-KEEPER 128

Time-Keeper may be used with almost any Basic program, including programs published in *RUN*. Time-Keeper is compatible with RUN Script 128, version 2.40, published in the December 1986 issue and on the Productivity Pak II disk. It also works with version 2.56 on the Productivity Pak III disk as long as the spelling checker is not activated. Unfortunately, the spelling checker may use the memory occupied by Time-Keeper and lock up the computer.

Also, RUN Script and other word processors continually update the 80-column screen, and, because of this, a conflict can result over control of the VDC chip. For this reason, the clock display must be off before you load and run the word processor.

To view the clock/calendar display from within RUN Script, press F1 followed by shift/control/N. Turn the display off again with F1 followed by shift/control/F. If you leave the continuous display on, garbage can appear on the first line of your text display. These characters are not stored in RAM and will not affect your document in any way. The original text can always be returned by pressing

the restore key, but it's best to view the display and then turn it off again.

Time-Keeper has also been tested with Notepad 128, Presto-Write 128, Shopping List, SpeedScript 128, and the Merlin 128 assembler from Roger Wagner Publishing. You'll find that it will also work with almost any Basic program. In most instances, Time-Keeper should be the very first program you load and run. When loading Notepad 128, however, don't run the boot program. Instead, load the object file with the Boot command. Since Time-Keeper uses the same time-of-day clock, it's not necessary to reset the time for Notepad.

FOR 128 PROGRAMMERS

The following RAM locations contain the data in the format indicated:

LOCATION	DATA	FORMAT
6410	tenths	BCD
6411	seconds	BCD
6412	minutes	BCD
6413	hours	BCD
6414	month	decimal
6415	date	decimal
6416	year	decimal

Binary Coded Decimal data (BCD) can be converted easily by using the 128's built-in hex\$ command. Use the following routine to read and print the time from within one of your programs:

```
100 FOR I=0 TO 2:B=PEEK(6413-I) AND 127
110 PRINT RIGHT$(HEX$(B),2);
120 IF I<2 THEN PRINT ":";
130 NEXT
```

Use the following routine to read and print the current date:

```
150 FOR I=0 TO 2:B=PEEK(6414+I)
160 PRINT MID$(STR$(B),2);
170 IF I<2 THEN PRINT"/";
180 NEXT
```

INTERRUPTING THE VDC CHIP

The *C-128 Programmer's Reference Guide* states that you shouldn't try to write to the 80-column chip either directly or indirectly by way of interrupts. Generally this statement is correct, but you can

work around it with a little ingenuity. Since there is no way to safely save the value of the VDC registers during interrupts, the best thing to do is not disturb them while they are in use. It's too bad that the operating system doesn't set a flag to signal when an important 8563 routine is active, but that would have made things too easy.

The first thing you can do is limit the number of accesses to the chip. The IRQ hits about 60 times a second, so your routine would be bombarding the chip with job requests. A built-in time delay can solve this problem easily by limiting the number of screen updates. Time-Keeper uses a one-second delay in its interrupt routine as the first trick.

The next trick is to check to see whether the Basic screen editor is using the display chip at the present time, and to exit the routine if it is. This can be established by examining the last JSR address on the stack and see if it falls within the range of one of the screen editor's ROM routines. If you write your own programs that access the 80-column chip, it is wise to use the built-in ROM routines rather than your own.

RUN it right: C-64; C-128 (40- and 80-Column modes)

Display Frame Animation

By Joseph R. Charnetski

BECOME THE ENVY of budding and experienced programmers alike with this title animation program. By rapidly cycling through a series of character colors, Display Frame Animation makes your menus and introductions come alive with neon screen animation rarely seen outside of commercial entertainment software.

As written, Display Frame Animation displays a default greeting, but you can substitute your own words in variable FL\$(x), where x is the number of lines down from the top of the display title.

While the default colors in line 60060 already create a colorful animated border, you can substitute red, orange and yellow to produce a glowing neon effect. To increase or decrease the speed of color cycling, change the value of FZ in line 60230.

You can add Display Frame Animation as a subroutine to your own programs. Just follow these steps:

1. Load your program into the computer.
2. In Direct mode, type:

```
L = PEEK(45) + PEEK(46)*256 - 2:POKE44,L/256:POKE43,L -  
PEEK(44)*256 <return>
```

3. Now load Display Frame Animation and enter:
POKE43,1:POKE44,8 <return>

4. Finally, remove the REM in front of the Return statement in line 60250.

RUN it right: C-64

Window Construction Set

By John Ryan

POP-UP WINDOWS AND MENUS are often used in a program to spice up what would otherwise be drab displays. Designing windows, however, can be a tedious and time-consuming task—even for the advanced programmer. My program, Window Construction Set (WCS), makes the job easy and offers several advantages over other window utilities currently available.

First and foremost, it requires no programming. All you need to do to create a colorful, three-dimensional, pop-up window is to choose the window's style and color, type in your text and then specify the location where you want the window to be on the screen. The computer does all the rest.

You can use WCS to creatively display everything from menus to program instructions, and blank windows are great for user input. Also, by placing several windows in the same screen position, you can rapidly "flip" through a lot of information without having windows scattered all over the screen.

WCS is a full-featured, menu-driven editor that's written entirely in machine language for fast and efficient operation. You can build several files, saving up to nine windows in each file, so you'll have

a wide choice to use in your own programs.

PRELIMINARIES

Load the file WCS to place the editor in memory. Then you can access it by typing SYS 49152, which brings up the main menu.

If you're creating a new window file, you must first select the window type—plain or bordered—and the color, which determines not only the color of the window body and border, but also of the text. Press T to toggle the window type and C to select color. Then press N to choose the window number (1-9) you want to edit.

THE EDITOR

The edit option is for creating new windows only. I'll describe later how to revise existing windows.

Press E to edit the window you selected. This brings to the screen a blank text area, measuring 16 columns across by 10 rows down and sporting the characteristics you assigned. This area contains a flashing cursor, and, as you'll see after you enter some text, it's surrounded by a one-row/one-column frame, for a total window area of 18 columns by 12 rows.

Type your text, positioned with the cursor keys, within the window. The return key is reserved for processing the window, so don't press return at this time.

The editor won't let you type past the window borders, and persistently trying to do so may corrupt part of a border. However, don't worry, the corruption won't show up in the final product.

The insert/delete key is disabled in the editor, as is the shift/return combination. You should also avoid pressing clear-home, or you'll erase the editing screen and have to press return to start again.

Pressing return processes your new window and takes you back to the main menu. There you can repeat the editing process to create as many windows as you need.

POSITIONING YOUR CREATIONS

Don't save your file yet; you still have to fix the position of your windows on the screen so the computer will know where to place them in your own programs. Press N to select a window and P for positioning, then use the cursor keys to place the small, black cursor where the upper-left corner of the window should be. The editor restricts cursor movement to an area in the upper-left region of the screen that's determined by the 18-column by 12-row size of the

window and the fact that the cursor represents the upper-left corner of the window.

When you've found a location that looks good, press the return key to fix it and make the window appear. If the placement doesn't look right, press P again and move the cursor to another spot (the editor remembers only the most recent). When that window is all set, position the rest of your windows by cycling between the N and P keys; then save the window file and press the Commodore key to erase the screen and return to the main menu.

SAVING AND REVISING

Press the F key to begin the save sequence; then, at the prompt, enter a filename of 16 characters or less. The Save and Load commands default to device 8, but the device number can be changed before you enter the editor with a POKE 51539, <device number>.

The revise option works exactly like the edit option, except it's for use on windows you've already created. Remember, pressing E completely erases any work you've done on a window, while R preserves your efforts. To load a WCS file so you can work on windows you've saved, press the L key and respond to the prompt.

When your windows are all set, but before including them in your own program, you should test them out with the small demo program, WINDOWS LIST2. Load the demo, enter a WCS filename at the prompt, then press the space bar to cycle through the windows. If they don't look right, reaccess the editor, load the file, revise or replace the windows, resave the file and test it again.

USING WINDOWS

To include the windows in a program of your own, insert the following lines at the beginning of your Basic listing:

```
10 IF FLAG = 1 THEN GOTO 40
20 IF FLAG = 2 THEN 50
30 FLAG = 1:LOAD "WCS",8,1 :REM LOAD IN MAIN PROGRAM
40 FLAG = 2:LOAD "<filename>",8,1 :REM LOAD IN YOUR FILE
50 REM YOUR PROGRAM GOES HERE
```

If you need more than nine windows, just add a line between lines 20 and 30 with the entry IF FLAG = 3 THEN GOTO <line number>; then follow the load format in line 30.

To call a window to the screen, use

POKE 820, <window number (1-9)>: SYS 49155

Whenever a window is called, WCS preserves the current screen in memory, to be restored with

POKE 820,0: SYS 49155

when you're ready to move on. You can study the demo program to see how to use this statement.

Note that you mustn't press the run-stop/restore combination before the screen has been restored, because it would confuse the pointer that keeps track of screen restoration. If your program needs to move on without restoring the screen, use the statement

POKE 821,0

This initializes the pointer, so the next time a window is called, the program will still know which screen should be restored.

PROGRAMMING NOTES

Machine language programmers must resort to the Kernal load routines to access WCS files. WCS occupies memory from \$C000 to \$C954. It also uses storage memory from \$CA00 to \$CFD2 and Basic ROM memory from \$B000 to \$B800 for screen and color RAM storage. (The window storage area is initialized each time you access the editor via SYS 49152.) Basic ROM is switched out periodically, but this shouldn't present a problem to machine language programs as long as WCS is used as a subroutine. If you plan to incorporate WCS into any development programs, be aware that the main editor steals the IRQ vector at \$0314.

RUN it right: C-64

Fontastic 64

By Kenny Lawson

DO YOU FEEL STIFLED by the lack of text fonts available for 64 Basic? Do you wish you could add creativity to your video screen? Well, this "fontastic" program for the 64 will let you do just that.

It gives you a choice of nine different text fonts to replace the

blah-looking set that comes with your computer. For example, you can choose underlined, cursive, all-caps, computer-style, thin-lettered and other texts.

FONT DESCRIPTIONS

Blockfont—changes all of the normal text letters to block letters, without affecting the graphics.

Capsfont—gives you all capital letters in large and small sizes.

Compufont—changes letters and numbers to early-computer-style letters, like the numerals you see on your bank checks.

Cursifont—displays an upper- and lowercase set of letters in cursive handwriting, but you lose some of your graphics symbols, just as with Capsfont.

Diamondset—a set of letters I designed to look diamond-shaped.

Dotfont—uses dots to form the letters.

Stencilfont—creates letters and numbers that look like they were drawn with an old-fashioned stencil, like the B on the Boston Red Sox caps (Go Sox!).

Thinfont—creates letters, numbers and some symbols that are only one pixel wide in most dimensions.

Underfont—creates an entire set of alphanumeric and symbols (including spaces) that are underlined.

All the fonts also support reversed characters.

USING THE PROGRAMS

To copy these fonts to another disk, load and run CHAR SET GENRTR. Next, load and run each character file that ends in the word "set." Each program opens a file to disk with its own filename

Figure 1. Examples of the fonts.

THIS IS BLOCK FONT.	THIS IS DOT FONT.
THIS IS CAPS FONT.	THIS IS STENCIL FONT.
THIS IS COMPU FONT.	THIS IS THIN FONT.
THIS IS CURSIVE FONT.	<u>THIS IS UNDER FONT.</u>
THIS IS DIAMOND FONT.	

and writes either data items or character ROM peeks to the file. Then it closes the file and resets the disk drive.

To use your font disk, which now has nine binary files containing nine new character sets, first reset your computer by turning it off and back on again. Load BOOT CHAR SET, which automatically loads FONT MENU. Once this is done, the menu page will appear on the screen. Use the cursor up/down key to highlight your choice of fonts and press the return key to load that font.

The screen will clear and the familiar cold-start screen (in the new font you selected) will appear. Note that you now have 36861 Basic bytes of memory available. You can load, save and run just about any program, except programs such as some word processors, spreadsheets, etc., that use locations 2048 to 4095. Also, if you have to hit the run-stop/restore key, just enter, in Immediate mode, SYS 755 to restore your new character set.

Now you are ready to use these new fonts in your own programs. Just be sure to save these font programs on your programming disks.

With this background on custom character sets, try designing and creating your own fonts.

RUN it right: C-128

Creating Windows

By J. C. Vollmer

THIS ARTICLE BRINGS YOU some useful techniques for manipulating windows on the C-128. The 128's window statement is controlled by five parameters and has the following syntax:

WINDOW <top-left col,top-left row,bottom-right col,bottom-right row,
clear flag>

To define a window, you must provide the coordinates of the top-left and bottom-right corners, in terms of the row and column numbers of those two points.

The column values must be in the range of 0-39 in 40-Column mode, and 0-79 in 80-Column mode. The row values must be in the range 0-23 for both modes. The optional clear flag blanks out the

window when the flag is set to the value of 1.

You can also create a window by moving the cursor to the desired top-left corner of the window and pressing the escape and T keys in sequence. Then move the cursor to where you want the lower-right corner and press the escape and B keys in sequence.

To clear the window, type in the following line:

```
WINDOW 0,0 RWINDOW(2) - 1,23,0
```

WINDOW DEMO is a demo for creating windows on both the 40- and 80-column screens. It's written for the 80-column screen, but will work in 40-Column mode if you change line 30 to the following:

```
30 UC = INT(RND(0)*20) + 1:LC = 37 - INT(RND(0)*18)
```

If you use your C-128 in 80-Column mode, try adding the lines below and see how they alter the windows:

```
85 IFINT(RND(0) + .5) THEN PRINT CHR$(14)::ELSE PRINT CHR$(142);
```

```
86 IFINT(RND(0) + .5) THEN PRINT CHR$(15)::ELSE PRINT CHR$(143);
```

```
87 IFINT(RND(0) + .5) THEN PRINT CHR$(2)::ELSE PRINT CHR$(130);
```

Line 85 toggles between Uppercase/Graphics mode and Upper-Lowercase mode. Line 86 toggles the 8563 chip's flash feature, which, in this case, lets you set one window to flashing while another remains normal. Line 87 turns the underlining feature on and off. The latter two lines have no effect in 40-Column mode.

BORDERS AND TITLES

Lines 6000-6070 of the program create titles and borders for the windows. You can use them as a subroutine to give you windowing control. Eight variables within these lines define the following parameters:

UC—Upper-left column

UR—Upper-left row

LC—Lower-right column

LR—Lower-right row

CL—Clear flag

BC—Border color

B\$—Border character

T\$—Title of window

When defining a window with this subroutine, leave space between the window and the edge of the screen for the window's border, and

set T\$ to null (T\$ = " ") if you don't want to have a title. Also, make sure the title is no longer than the width of the window or it'll get truncated.

RUN it right: C-128 (40- or 80-Column mode)

Cursor Commotion 128

By Jim Borden

WHEN YOU RUN CRSR COMM. 128, it checks to see whether you are creating a new file or reading an existing one back. You can press any key but F7 in creating a document. Use F7 only to signal that the document is done.

Because the program uses F7, you must enter POKE 828,183 before typing it in. This redefines the function keys to the C-64 values. To reinstate the default C-128 values, use POKE 828,173.

I suggest that you take a look at CRSR COMM. DEMO to get an idea as to how you can use the program.

RUN it right: C-64

Video Flashcards

By Kevin B. Alton

VIDEO FLASHCARDS IS AN EDUCATIONAL practice game for children who are learning the fundamentals of addition and subtraction. Written for the C-64, this Basic 2.0 program features a simple and easy-to-read menu-driven routine. Students of varying skill levels will be attracted to the colorful, oversized numbers and the musical fanfare that follows all the problems that are correctly answered.

GETTING STARTED

The opening menu of Video Flashcards determines the skill level of each drill. Press return and the program begins using the default values. To modify the menu, the space bar moves the pointer to any line. The values can then be changed by pressing +, - or *.

The maximum number that you enter in response to the menu prompts determines the largest integer that can appear as an individual number, sum or difference. This parameter can be quickly changed from 1 to 9999 because the increment rate accelerates as the value becomes larger.

The length of a practice session varies from 1 to 100 problems. The * toggles between Addition and Subtraction modes. If you choose both types simultaneously, they'll be randomly incorporated into each drill.

You can also adjust the degree of difficulty by increasing or decreasing the time allowed for answering problems. Valid times range from one second to one minute. After every drill, you can return to the menu or exit the program.

ADDITIONAL FEATURES

Answers for each problem are input from right to left—just as if written on paper—and entered with a return. Correct mistakes with the insert-delete key.

A timer appears near the home position of the screen and displays, with a yellow inverse video, the seconds accumulated for each problem. Only three seconds remain when it turns red.

A wrong answer or time-out results in a second chance; otherwise, the next problem is automatically displayed. If all problems in each drill are answered correctly on the first try, your "prize student" is treated to a short fanfare.

PROGRAM FOOTNOTES

Every effort has been made to "kiddie-proof" this program from accepting invalid entries. Consultation with my nine-year-old son, who is an expert at crashing programs, proved to be invaluable.

The shift/Commodore keys are also disabled, because all the graphically generated numbers require Uppercase mode. This is accomplished in line 10 with a Print CHR\$(8). The input section (lines 420-500), which updates the current menu display, takes advantage of the number momentarily stored in memory location 197 during

the action of pressing any key. The returned value, Peek(197), is the keyboard code that the operating system ultimately converts to an internal character code and its ASCII representation. If no key is pressed, Peek(197) returns a value of 64.

Line 420 continually polls the keyboard until a valid key (+, -, *, space or return) is pressed. This approach to dynamic input is an attractive alternative to the Get statement when only a few keys are needed for program control. The input routine (lines 850-920) for each addition or subtraction problem is more conventionally controlled by the Get statement.

The only valid responses allowed are the numbers 0-9, delete or return. A careful examination of lines 870 and 880 also reveals that the delete and return keys are only effective when at least one number has been displayed in the answer.

Last but not least, the default values for the menu, located in line 410, are assigned to the variables MAXSIZE, NUMBER, ADD\$, SUBTRACT\$ and SEC. They can be adjusted to more closely match the individual skills of your children. I usually maintain a personalized copy for each of my children. That helps to simplify the start of a practice session.

RUN it right: C-64

C-64 Command Collector

By William Mark

IF YOU MUST PERFORM computer tasks that involve entering the same series of commands over and over again, a batch processing program that does this job for you can be a real time and aggravation saver. The concept is simple. You place the sequence of commands in a special file on disk; then, when you enter the single command to start the batch processor, it executes the commands in the file on its own. It resides in memory as a wedge, staying out of the way until it's needed, so you can use your computer normally.

Batch processing is a handy feature, but unfortunately the C-64 doesn't come with it built in. I wrote the Command Collector pro-

gram to help to compensate for this shortcoming.

THE COMMAND FILE

Before loading COMMAND COLLECT into memory, you must create the file containing the commands to be executed. This can be done with almost any word processor or utility that creates the sequential files, because the program can read most text files, as long as they're free of formatting commands.

Enter each instruction exactly as you would from Basic. You can include a number of instructions, separated by colons, in a line, but it's easier to edit them if you use one command per line.

THE PROGRAM

COMMAND COLLECT is a boot routine that activates the main program when you call it via the SYS command mentioned below. A second routine is the main routine, which holds the commands that come from the disk file.

A third small routine is needed to call the main program if the latter resides in its default location, which is under Basic ROM. Finally, a fourth routine is a zero-page, two-byte pointer for indexed, indirect addressing by Command Collector's machine language code.

To activate Command Collector, just type SYS 828, its default location, or whatever other address you may have chosen. The program takes over the operation of your C-64, calling up commands from the disk file as if you were typing them in. When the final command has been executed, the program deactivates itself and returns control to the keyboard.

USING COMMAND COLLECTOR

When you run Command Collector, it asks you for the memory addresses where you want it installed. If the default values are all right, just press the return key at each prompt. The default addresses will work in most cases, but there may be some programs that occupy the same addresses as Command Collector. Such a conflict would cause the computer to crash and if this happens, you must change Command Collector's memory locations. You can also change the highest allowable end address for the main routine, to prevent overflow from a command file that's too large.

The starting addresses for all the routines can be altered as well. The main routine can go under Basic ROM (\$A000-\$BFFF), and the other routines can go in the Basic program area above the main

routine. However, bear in mind that Basic programs may overwrite this area. To prevent overwriting, reset the top-of-memory pointer with POKE 55,LOW:POKE 56,HIGH:CLR.

Addresses can be input in decimal or hexadecimal (preceded by \$). As a safety precaution, an error message appears if you enter an invalid address for any of the routines.

After you've specified addresses for the routines, the program asks for the name of the command file, its filetype (sequential, program or user) and its format (screen-code or Commodore ASCII).

If you use screen-code files, Command Collector offers a special convenience. Reverse-field @ - Z and reverse-field 1-8 (Poke codes 128-154 and 177-184) are interpreted as control/@-control/Z and F1-F8, respectively. This feature is handy for batch execution of programs that use the control key function keys.

Once you've specified the filetype and format, the program reads the command file into memory, then displays the SYS address (828, unless you change the defaults), so you can start the processing. As long as the routines remain undisturbed in memory, you can call Command Collector repeatedly by entering the same SYS command.

After you've entered the SYS command, the boot routine wedges the main routine into the C-64's interrupt routine. If the main routine is hidden under Basic, as it is by default, Basic must be switched out before the program runs and switched back in afterward. It's the third routine that performs this switching.

The main routine checks to see whether the keyboard buffer is empty. If so, it puts a character from the command file into it.

USES FOR COMMAND COLLECTOR

Command Collector has unlimited potential. One possible application is in running assemblers, such as the Commodore Macro Assembler, since the same sequence of commands is used each time you assemble a program. Command Collector can also be used to merge Basic programs by placing a needed subroutine in a command file as follows:

```
REM WRITE SUBROUTINE LISTING TO DISK AS SEQUENTIAL FILE
LOAD "SUBROUTINE",8
OPEN 8,8,"SUBROUTINE.ASCI,8,W"
CMD 8
LIST
PRINT #8
CLOSE 8
```

Then run Command Collector as follows, using SUBROUTINE .ASCII as the command file:

```
LOAD "PROGRAM",8  
SYS 828  
SAVE "NEWPROGRAM",8  
REM "NEWPROGRAM" INCLUDES THE SUBROUTINE
```

Let your imagination run free, and you're sure to find other applications for Command Collector.

RUN it right: C-64; printer

Mister Poster

By Frank Colaricci

MISTER POSTER IS a machine language program that lets you preview, in black and white on a screen, the bitmap part of a picture program file, then print the screen image as spaces and asterisks in 80-column sections on your printer. It will load the bitmap part of picture files generated by the following graphics packages: Blazing Paddles (Baudville), Cadpak-64 (Abacus), Chartpak-64 (Abacus), Doodle! (Crystal Rose), Koala (Koala Technologies) and The Print Shop (Broderbund). It can also load bitmaps that have been saved as program files with any of the following load addresses: \$2000 (8192), \$4000 (16384), \$6000 (24576), \$A000 (40960) and \$E000 (57344).

I've tested Mister Poster on a system combining a C-64, a Cardco G-Wiz interface with all switches off and a Star SG-10 printer with all switches on. It also works with a Commodore MPS-801 printer when using a secondary address of zero.

Load MR.POSTER.GEN from the ReRUN disk, and save a backup copy to a work disk. Then run the program. The Basic machine language generator takes about 1½ minutes to write an 8-block machine language program named MR.POSTER.RUN to disk.

Table 1 provides a list of Poke addresses, present values and functions that you may want to use to adapt Mister Poster to your computer system. To make changes, load MR.POSTER.RUN, poke in the values you want in Direct mode (no line numbers, please),

then save the program with a different filename.

USING MISTER POSTER

When you run Mister Poster, a welcome message appears, along with a prompt to insert a write-protected disk containing the picture program files you want to print. (Mister Poster shouldn't write to your disk, but write protection is just good practice.)

After inserting the disk, press return to search its directory for program files that are larger than 31 blocks and smaller than 42 blocks. If there aren't any, a message to that effect appears on the screen. Then you can press any key to return to the first screen, insert another disk and press return, or you can exit the program by pressing the stop key.

If picture program files that might contain bitmaps are found, a list appears on the screen. Select the one you want by entering the letter code to the left of the filename. If it turns out to be a file, such as a Basic program, that Mister Poster doesn't recognize, a message to that effect appears, and you can press any key to return to the list of filenames.

If the load is successful, you're greeted by the following options:

F1—View bitmap

F3—Negative bitmap

F5—Print 80-column markers (< >)

F7—Print poster

Stop key—Quit

Table 1. Pokes you can change to adapt Mister Poster to your computer system.

Poke Address	Present Value	Function
2087	4	Printer device code.
2088	4	Printer secondary address. (Users of Commodore or third-party printers, such as the Star SG-10C or Gemini II, with built-in Commodore interfaces, should set this location to 0.)
2089	8	Disk device code.
2090	66	Linefeeds per page.
2091	32 " "	ASCII value of character printed when pixel is off.
2092	42 "*"	ASCII value of character printed when pixel is on.

Hold down F1 to view the bitmap in black and white and release it to return to the menu. Note that some Multicolor-mode files, such as those from Koala, may take on a different appearance in Black-and-White, Hi-Res mode. The reason for this is that in Multicolor mode bitpairs determine the color that's displayed, while Mister Poster represents the unaltered bitmap with white dots where the pixels are off and black dots where the pixels are on.

Hold down F3 to view a negative image of the bitmap and release it to return to the menu.

Press F5 to print a marker, consisting of a < and a > at either end of an 80-column line, to help you align the paper horizontally in your printer. It's important to align the paper, because the poster is printed in four sections, and it's easier to overlay the sections if column 1 is next to the left perforation or column 80 is next to the right perforation. When you've completed the alignment, adjust the paper to the top of the next sheet.

Press F7 to print the poster. The bitmap is visible on-screen during printing, and the pixels go negative after they're scanned, so you can monitor the program's progress.

Press the stop key to exit Mister Poster. You'll have a chance to reconsider, and if you decide to print another poster, just make sure the correct disk is in the drive, type RUN and press return. During printing, hold down the stop key to interrupt the output; then press F1 to continue printing or F3 to return to the previous menu.

ASSEMBLING YOUR POSTER

Assuming that you use 11-inch paper and 66 linefeeds per page, Mister Poster should take 16 sheets of paper to make your poster, with each of the four sections starting at the top of a new sheet. At the beginning and end of each 200-line section, you'll find a < printed in the first column and a > printed in the last column.

When the printing is done, tear the sprocket-hole strips off all the sheets and separate the printout at the beginning of each section. Then overlap the four sections and line up the <> markers. Finally, tape the sections together with cellophane tape, and, *voila*, there's your poster! ■

RE **== RUN**

TECHNICAL EDITOR
TIM WALSH

MANAGING EDITOR
SWAIN PRATT

COPY EDITOR
PEG LePAGE

PROOFREADER
HAROLD BJORNSEN

DESIGN AND LAYOUT
ANNE DILLON

TYPESETTING
DEBRA DAVIES
BETH KROMMES
KEN SUTCLIFFE

RERUN MANAGER
PAUL FINCH

CUSTOMER SERVICE REPRESENTATIVES
DEBBIE BOURGAULT
ANNE JOHNSON

ReRUN back issues at only \$16.47 each*

— Jan./Feb. 1987
— Mar./Apr. 1987
— May/Jun. 1987
— Jul./Aug. 1987

— Sep./Oct. 1987
— Nov./Dec. 1987
— Productivity Pak II
— Jan./Feb. 1988
— Productivity Pak III

** Price includes disk, documentation booklet, postage and handling. For foreign air mail, please add U.S. \$3.95 per order. Prepayment required.*

☐ Payment Enclosed ☐ MC ☐ VISA ☐ AE

Card # _____ Exp. Date _____
Name _____
Address _____
City _____ State _____ Zip _____
Signature _____

ReRUN • 80 Elm Street • Peterborough, NH • 03458
Call TOLL FREE 1-800-343-0728

BEAT THE RUSH!

Please send me:

- ☐ 1 year (6 issues) for \$69.97
☐ May/June 1988 ReRUN disk for \$16.47.*

**Available in May 1988.*

Includes programs for C-64 and C-128 (in both 64 and 128 modes).

Price includes postage and handling. For foreign air mail, please add U.S. \$3.95 per item or \$23.70 per subscription. Prepayment only.

☐ Payment Enclosed ☐ MC ☐ VISA ☐ AE

Card # _____ Exp. Date _____
Name _____
Address _____
City _____ State _____ Zip _____
Signature _____

ReRUN • 80 Elm Street • Peterborough, NH • 03458
Call TOLL FREE 1-800-343-0728